28

Enciclopedia Práctica de la

INFORMATICA

MEMORIAS A SEMICONDUCTORES / HARDWARE: **JUPITER ACE**

PROGRAMACION MODULAR / PLOTTERS BENSON

ORDENADORES SOBRE RUEDAS





Nueva Lente / Ingelek

125 ptas.

MEMORIAS A SEMICONDUCTORES

LAS unidades de memoria central en los sistemas microordenadores son memorias integradas a semiconductores. Las tradicionales unidades de almacenamiento a núcleos de ferrita o hilos plateados han perdido absolutamente su vigencia.

El usuario de microprocesadores debe familiarizarse con observar a las unidades de memoria bajo el prisma de simple circuito integrado (chip) de 16, 18 ó 24 patillas.

La evolución tecnológica en cuanto a densidad de integración crece día a día. En la actualidad se encuentran en el mercado memorias RAM y ROM integradas capaces de almacenar 16 Kbytes, 32 Kbytes e incluso 64 Kbytes.

A partir de la descripción anteriormente efectuada acerca de las memorias RAM y ROM, el lector puede deducir fácilmente que la base tecnológica de ambos tipos de memoria será obviamente distinta. Cabe indicar que dada la diversidad de técnicas empleadas en la fabricación de memorias integradas a semiconductores nos vemos en la necesidad de concretar nuestro estudio en aquellas técnicas de mayor importancia y difusión.

Tecnología de las memorias RAM a semiconductores

La mayor parte de memorias integradas RAM a semiconductores se hallan confeccionadas basándose en las siguientes tecnologías:

RAMs unipolares: Tecnología NMOS, VMOS y CMOS/SOS.
RAMs bipolares: Tecnología TTL.

A pesar de las favorables perspectivas de futuro lo que permite augurar el notable avance de la VMOS, la tecnología dominante en la actualidad en el campo de las RAMs a semiconductores es la NMOS. Debido a ello, centraremos nuestro estudio en las memorias basadas en la tecnología unipolar NMOS.

Las memorias de lectura/escritura RAM se catalogan de acuerdo a dos tipos perfectamente diferenciados: Memorias RAM estáticas o dinámicas.

La naturaleza de estática o dinámica es conferida a la memoria por la estructura del punto básico de almacenamiento. Así pues, una memoria RAM será de uno u otro tipo según esté constituida por puntos de memoria estática o puntos de memoria dinámica.

Memorias RAM estáticas

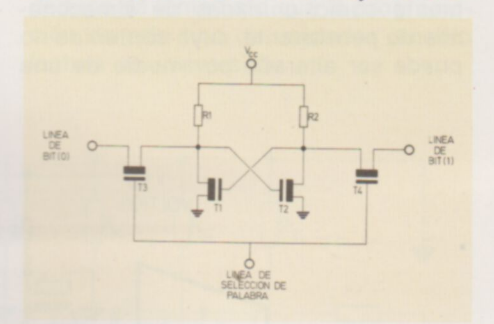
Los puntos de memoria de una RAM estática son elementos biestables que retienen la información almacenada, hasta que ésta es modificada por una nueva operación de escritura o hasta que se desconecta la alimentación.

La estructura típica de una memoria RAM estática coincide con la que se ilustra en la figura correspondiente. Los transistores MOS canal n se hallan acoplados conformando un circuito biestable o flip-flop capaz de almacenar un elemento de información binaria: 0 ó 1.

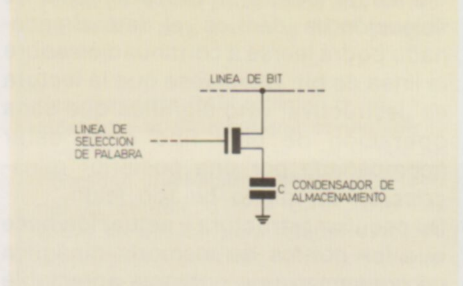
El mayor inconveniente de las memorias RAM estáticas lo constituye su elevado consumo energético. Ello se debe a que las resistencias de carga R_1 y R_2 consumen permanentemente al mantener el estado lógico en el que se halla posicionado el biestable. Esta deficiencia es subsanada parcialmente en las nuevas RAMs estáticas, empleando resistencias de polisilicio de elevada magnitud.

Para leer la información almacenada en el punto de memoria se introduce un pulso de tensión a través de la línea de selección, lo que provocará una corriente a través de la rama T_1-T_3 o T_2-T_4 según que el bit almacenado sea 0 ó 1. En definitiva, la lectura se efectúa detectando la presencia de corriente en una u otra línea de bit.

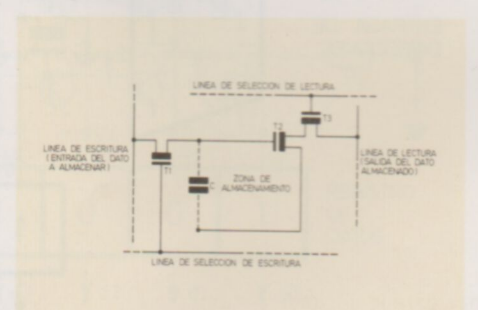
Se observa que cada punto de memoria RAM estática incorpora un notable número de elementos. Esta circunstancia limita las posibilidades de integración de este tipo de unidades, de tal forma que los chips comerciales no suelen



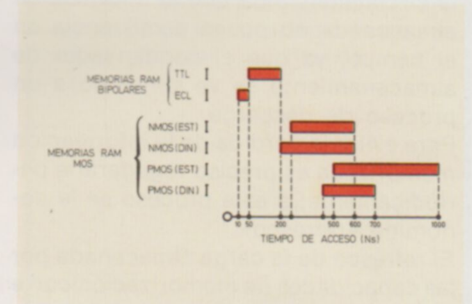
Punto de memoria de una RAM estática, basado en cuatro transistores MOS. El estado de conducción o no conducción de cada uno de ellos se interpreta como un «1» o un «0» lógico, respectivamente.



Estructura circuital básica de un punto de memoria RAM dinámica a transistores MOS. En este caso el condensador almacena, o no, una determinada carga eléctrica que representa a un «1» lógico.



Punto de memoria de una RAM dinámica basado en tres elementos MOS. La zona de almacenamiento de carga, constituida por un condensador, es quien mantiene al circuito en un estado lógico bajo o alto. El transmisor T1 se encarga de gestionar las operaciones de lectura.



Tiempo de acceso correspondiente a las RAMs a semiconductores. El margen de variación asignado tiene en cuenta los diversos tipos de RAMs clasificados dentro de una misma familia tecnológica.

MEMORIAS A SEMICONDUCTORES

superar una capacidad de almacenamiento de 4 Kbytes (4.096 bits).

Memorias RAM dinámicas

Los puntos de memoria dinámica almacenan la información en forma de carga, o ausencia de carga, en un condensador.

Para entender el funcionamiento de un punto de memoria dinámica analizaremos el circuito que aparece en la figura correspondiente.

Las operaciones de lectura o escritura se desencadenan al mandar un pulso a través de la línea de selección. Cuando la operación es de escritura, debe actuarse sobre la línea de bit cargando el condensador —almacenamiento de un estado lógico «1»— o inhibiendo la carga si se desea memorizar un «0» lógico. Por lo que respecta a las operaciones de lectura, el procedimiento de selección es idéntico; el dato almacenado podrá leerse a continuación sobre la línea de bit. Obsérvese que la lectura es destructiva; esto significa que cada operación de este tipo debe estar acompañada por una posterior reinscripción del dígito binario leído.

Su peculiar estructura y actuación hace que los puntos de memoria dinámica no consuman una potencia apreciable para mantener la información almacenada. La disipación es significativa únicamente en el instante de proceder a una lectura o escritura.

El bajo consumo y la simplicidad circuital son factores notablemente positivos, no obstante, cabe considerar aún la circunstancia de que la información almacenada no posee persistencia en el tiempo, ya que el condensador de almacenamiento se ve sometido a un proceso de descarga.

Para evitar la pérdida de la información memorizada es preciso regenerarla periódicamente; a este proceso se le denomina «refresco».

El refresco de la carga almacenada por las capacidades de memorización corre a cargo de una señal de control pulsatoria, cuya periodicidad es del orden de 2 ms en los chips comerciales.

En definitiva, las características más importantes de las memorias RAM dinámicas se resumen en el mínimo consumo energético necesario para retener la información y en el reducido número de componentes que constituyen cada punto de memoria. Esta última característica facilita las posibilidades de integración de los puntos de memoria y, en consecuencia, amplía el nivel de integración de las RAMs dinámicas y reduce su coste respecto al de las RAMs estáticas. En cuanto a los factores que determinan la conveniencia de utilizar uno u otro tipo de memoria RAM, diremos que las RAMs dinámicas se emplean cuando la capacidad de memoria necesaria es relativamente elevada. En el caso contrario es preferible optar por RAMs estáticas.

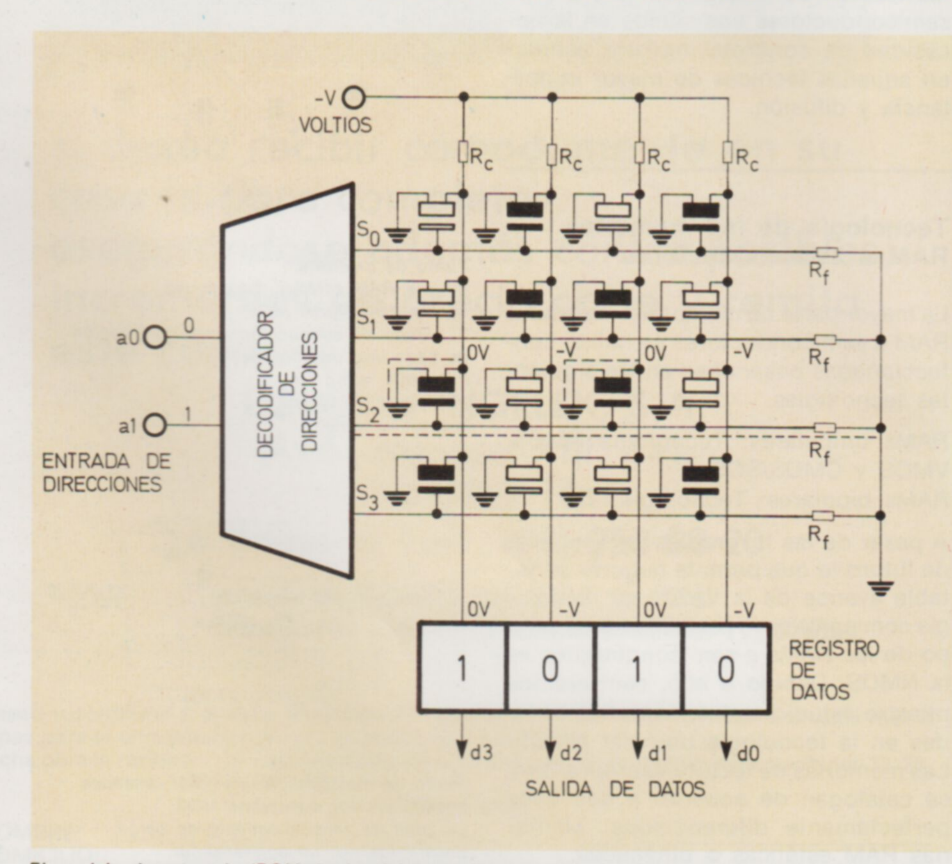
Tecnología de las memorias ROM

Las memorias ROM o memorias de sólo lectura han sido definidas anteriormente como unidades de almacenamiento permanente, cuyo contenido no puede ser alterado por medio de una operación convencional de escritura. La información se halla «grabada» sobre la propia estructura de la unidad de memoria, siendo, en consecuencia, inalterable.

La organización interna de las ROMs suele ser del tipo denominado «por palabras». Esto es, la memoria aparece como una matriz de tantas filas como palabras almacena, y tantas columnas como bits constituyen cada palabra.

Existen diversos tipos de ROMs. Estrictamente, el apelativo ROM corresponde a las memorias de sólo lectura, programadas por «máscara» durante el proceso de fabricación. Otro apartado lo constituyen las memorias de sólo lectura programables por el usuario: PROMS. Finalmente, están las memorias de sólo lectura, cuyo contenido es modificable por medio de procedimientos especiales: memorias EAROM, UV-EPROM...

Para ilustrar las diversas estructuras internas la mayor importancia en el campo de las memorias de sólo lectura, nos concretaremos en un modelo de



El modelo de memoria (ROM de 4 palabras × 4 bits) ilustra el desarrollo de una operación de lectura sobre la célula cuya dirección es 01.

memoria capaz de almacenar 4 palabras, dato de 4 bits cada una.

ROMs programadas por máscara

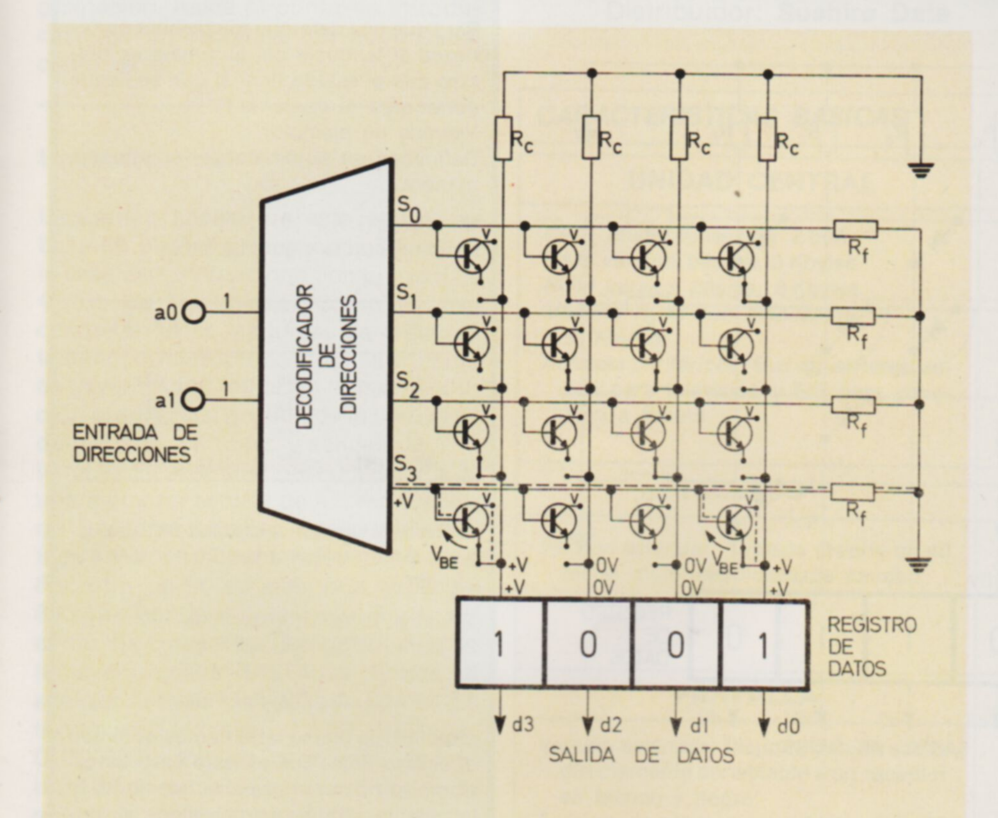
Recordemos que en este caso la información es grabada durante el proceso de fabricación. En el modelo ejemplo se observa que la información binaria: 1 ó 0 lógico, aparece en forma de presencia o ausencia de un diodo en los nudos de la matriz.
El decodificador de direcciones seleccionará una de las 4 células, cuyo contenido pasará al registro de información según las condiciones siguientes:

- En el nudo que no haya diodo se leerá un estado lógico alto.
- En el nudo sin diodo se leerá un «0» lógico.

La programación de las ROMs de este tipo se efectúa en el transcurso del proceso de fabricación, creando un diodo de acoplamiento en los puntos de memoria que deban almacenar un «1» lógico.

Estructura de las memorias PROM

Las PROMs (Programmable Read-Only Memory) permiten una única programación de origen, que puede realizar el propio usuario. Comúnmente, las PROMs son de tecnología bipolar y su programación se efectúa destruyendo los fusibles de acoplamiento en los nudos que deben memorizar un «0» lógico.
El fabricante entrega las PROMs con los acoplamientos intactos. Una vez decidida la información a almacenar, el usuario debe proceder a eliminar los acoplamientos que corresponda.
El procedimiento de trabajo de las memorias PROM es semejante al descrito para el caso de las ROMs. La programación se efectúa destruyendo el fusible en los acoplamientos o puntos de memoria en los que se desea grabar un «0» lógico. Observando la estructura elemental de cada punto de memoria se deduce de inmediato que la programación de este tipo de unidades de sólo lectura es irreversible.



Organización típica de una memoria ROM con acoplamiento a transistores unipolares MOS. Los transistores en blanco representan puntos de memoria creados para el almacenamiento de un «0» lógico. Los transistores en negro representan puntos de memoria para el almacenamiento de un «1» lógico.

Glosario

¿Qué ventajas presentan las memorias RAM dinámicas sobre las RAM estáticas?

La principal ventaja es el menor consumo energético de las dinámicas. Por otra parte, las posibilidades de integración de estas memorias también son mayores. Las RAMs estáticas se utilizan, en general, cuando la capacidad de memoria requerida no es muy alta.

¿Hay algún tipo de memorias ROM cuyo contenido pueda alterarse por el usuario?

Sí. Salvo las memorias ROMs programadas por máscara, cuyo contenido es grabado durante el proceso de fabricación, las memorias UV-EPROM, EAROM, etc., admiten cambios en la información almacenada.

¿Cuál es la diferencia entre una memoria RAM y una memoria ROM programable?

Las memorias RAM retienen la información sólo si se mantiene conectada la tensión que las alimenta. Las ROMs no necesitan una tensión exterior para retener el contenido. Por otra parte, las RAMs se utilizan para retener datos de un programa concreto, mientras que las ROMs se utilizan únicamente para almacenar datos internos de la máquina, no modificables durante la ejecución de un proceso particular.

¿Existen memorias ROMs dinámicas?

No. El contenido de las memorias ROMs no se desvanece con el tiempo y no es, por tanto, necesario aplicar ningún tipo de tensión de alimentación para conservar ni regenerar su contenido.

MEMORIAS A SEMICONDUCTORES

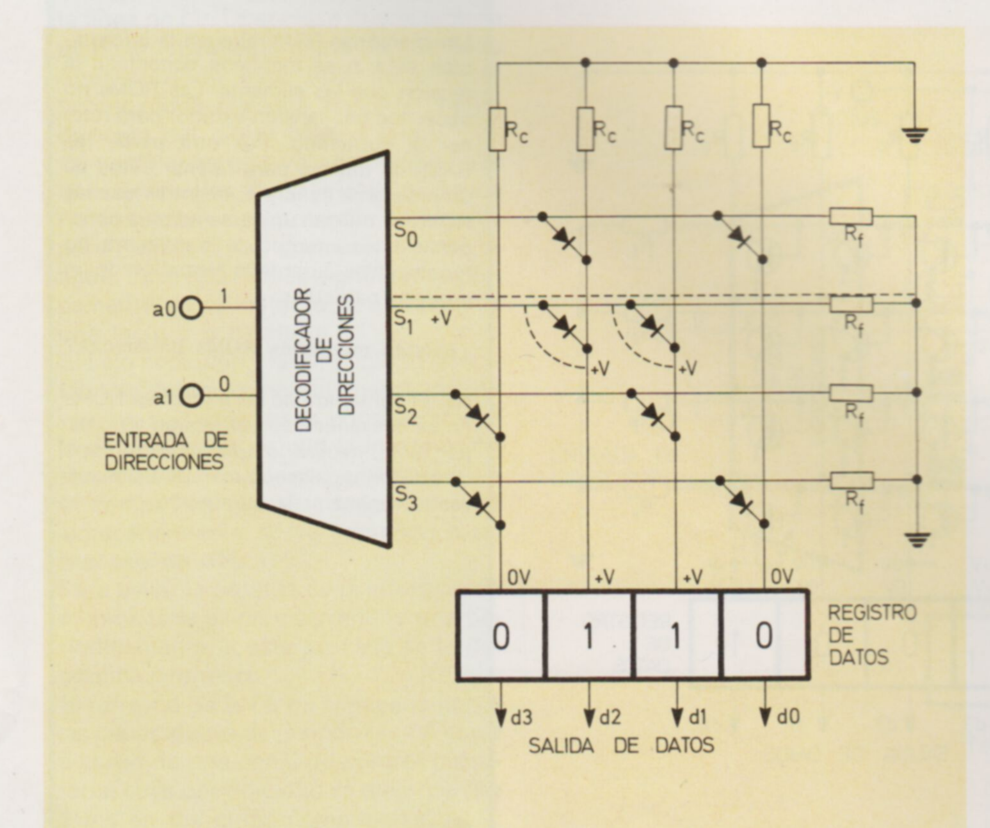
Estructura de las memorias UV-EPROM

Las UV-EPROM son memorias de sólo lectura en su modo operativo normal, programables eléctricamente por el usuario y que pueden ser borradas sometiéndolas a una radiación ultravioleta. Realmente, este tipo de memorias permiten sucesivas reprogramaciones, aunque hay que recordar que su comportamiento dentro de un sistema de proceso se reduce al propio de una memoria de sólo lectura.

La estructura microelectrónica de una memoria UV-EPROM revela su peculiar fundamento tecnológico. La zona central constituye una puerta flotante aislada eléctricamente del resto de la estructura MOS. Aplicando una tensión de aproximadamente 25 V entre la puerta de control y el drenador, la puerta flotante recibe una acumulación de carga eléctrica que la convierte en un efectivo punto de memoria. La carga almacenada en la zona intermedia memoriza un estado lógico alto de forma permanente, hasta que sea borrado por efecto de la radiación ultravioleta. Esta operación de borrado equivale a dotar al dióxido de silicio que rodea la puerta flotante de la conductividad suficiente como para disipar la carga almacenada.

Una vez programada la memoria, la información permanece inalterable, aun en el caso de efectuar sucesivas operaciones de lectura o desconectar las líneas de alimentación. Para reprogramar la memoria se recurre a la acción de la luz ultravioleta; a tal efecto, en la parte superior del chip existe una abertura que facilita el acceso de la radiación a la superficie del circuito. Para evitar un posible borrado accidental debe ocultarse la superficie semiconductora de la luz exterior durante su operación normal.

Otro tipo de memorias reprogramables de sólo lectura son las EAROM (Electrically Alterable Read-Only Memory). La estructura interna es similar a la de las UV-EPROM, con la salvedad de que el borrado y la programación se efectúan por medios exclusivamente eléctricos.



Organización típica de una memoria PROM a transistores bipolares con acoplamiento a fusible. Mientras éstos permanecen intactos, almacenan virtualmente un estado lógico «1». Cuando éste se rompe el estado lógico grabado es un «0».

Conceptos básicos

Reconocedores de lenguajes (y II)

Decíamos en el número anterior que uno de los métodos que usan los ordenadores para reconocer lenguajes es mediante autómatas. En este fascículo vamos a analizar este tipo de máquinas. La expresión formal de un autómata es la siguiente:

$A = \{Q, A, f, q, F\}$, donde:

- $Q = \{q_0, ... q_n\}$ es el conjunto de posibles estados del autómata.
- f es una función de transición $Q \times A \rightarrow Q$, que en función del estado en que se encuentre la máquina en un momento, y de la letra del alfabeto que se aplique lleva al autómata a un nuevo estado.
- q_0 es un elemento de Q llamado elemento inicial.
- F es un subconjunto de Q que contiene todos los elementos finales.

Para que una determinada palabra pertenezca al lenguaje del autómata es preciso que el estado final al que se llegue pertenezca al conjunto F.

Veamos un ejemplo:

Definimos un autómata de la siguiente manera:

$Q = \{q_0, q_1\}$, $A = \{0, 1\}$, $F = \{q_1\}$, y f está definida por la siguiente tabla:

X	Y	f(X, Y)
0	q_0	q_0
0	q_1	q_1
1	q_0	q_1
1	q_1	q_0

Queremos saber si la cadena 0110 pertenece al lenguaje reconocido por este autómata.

1.ª letra: 0, q_0: $f(0, q_0) = q_0$
2.ª letra: 1, q_0: $f(1, q_0) = q_1$
3.ª letra: 1, q_1: $f(1, q_1) = q_0$
4.ª letra: 0, q_0: $f(0, q_0) = q_0$

Al aplicar la última letra hemos obtenido el estado final, que en este caso es q_0. Como q_0 no pertenece al subconjunto F, la palabra «0110» no pertenece al lenguaje. Si, por el contrario, el último estado hubiera pertenecido a F, la palabra hubiera sido reconocida.

JUPITER ACE

LA multitud de pequeños ordenadores que ha inundado el mercado en poco tiempo, caracterizados por sus precios altamente competitivos y cada día más asequibles para el bolsillo del aficionado medio, suele tener una serie de rasgos muy comunes. Sin embargo, hay máquinas cuya acusada personalidad no se ajusta a los estándares establecidos: una de ellas es el Júpiter ACE. Este microordenador destaca tanto por el lenguaje con que trabaja, el Forth, como por la velocidad con que opera el microprocesador Z 80A, capaz, casi, de competir con microordenadores de categoría superior en capacidad de proceso y en precio.
El aspecto externo del Júpiter ACE no permite adivinar las prestaciones que la máquina esconde bajo su blanda carcasa de plástico blanco. La simplicidad del diseño exterior y la pobreza de detalles se ven compensadas por posibilidades de expansión interesantes, precio francamente barato y un indudable atractivo para los aficionados a la programación, hasta el punto de introducirse en un lenguaje tan poco usual como es el Forth.

Unidad central

La unidad central de este equipo se basa en el microprocesador Z 80A, de la casa Zilog, y funciona con un reloj de 4 MHz de frecuencia. Como es frecuente en estas máquinas de reducido tamaño, el microprocesador se encarga de llevar a cabo todas las tareas: ejecución programas, gestión de la pantalla y del teclado, entradas y salidas del sistema, etc. La versión básica dispone de una memoria central de acceso aleatorio RAM, de 3 Kbytes que pueden ser fácilmente ampliados hasta 51 Kbytes. Por otra parte, su memoria ROM de 8 Kbytes contiene el lenguaje Forth, así como los comandos de control del sistema, entre ellos VLIST, que permite visualizar la tabla de vocabulario, los de tratamiento de pilas SWAP, REPEAT, DUP, ROT... Incluye, asimismo, las instrucciones de cálculo, las utilidades de entrada y salida, los comandos del intérprete y del compilador, además del editor. Una instrucción especial, BEEP, permite la generación de sonidos por medio de su altavoz interior. Cada sonido queda definido por dos números que indican el tono y duración.
La comunicación con los periféricos se lleva a cabo a través de un bus externo de expansión, al que se accede por la parte posterior de la carcasa. Es posible la conexión de una amplia gama de dispositivos, entre ellos ciertos módulos destinados al Sinclair ZX 81, equipo con el que guarda alguna similitud. Incorpora, asimismo, un interface para casete y las tomas respectivas para una pantalla de televisión y la fuente de alimentación.

Teclado

El Júpiter ACE incorpora un teclado clásico QWERTY de teclas flotantes de caucho que, al igual que la carcasa, contribuyen al simpático aspecto de microordenador de juguete que presenta esta máquina. Pese a que el teclado consta únicamente de 40 teclas, dispone de un completo juego de caracteres que incluye símbolos particulares de programación («arroba», corchetes, llaves...) lo que resulta infrecuente en equipos pequeños. El teclado genera mayúsculas y minúsculas, que pueden ser redefinidas por el programador. Esto permite, por ejemplo, adaptar el teclado a las peculiaridades del idioma castellano.

Por otra parte, cada tecla es autorrepetitiva. Es posible el video inverso. La tecla «BREAK» puede detener la ejecución de un programa o salir del mismo en caso de bucle sin fin, pero el fabricante advierte, con honestidad, que este comando resulta a menudo inoperante. Ello se debe a que está controlado por programa, de modo que en determina-

Ordenador: **Júpiter ACE**
Fabricante: **Jupiter Cantab. Ltd.**
Nacionalidad: **Británica**
Distribuidor: **Sushiro Data**

CARACTERISTICAS BASICAS

UNIDAD CENTRAL	MEMORIAS DE MASA
CPU: Microprocesador Z 80A. *RAM versión básica:* 3 Kbytes. *ROM versión básica:* 8 Kbytes. *Máximo ROM con ampliación:* 51 Kbytes. *Accesos periféricos:* Bus del sistema, salida para televisión y E/S para conexión a casete.	El sistema dispone de E/S para la conexión directa de un magnetófono a casete de tipo convencional.
TECLADO	**LENGUAJES**
Versión estándar: Teclado Qwerty de 40 teclas multifuncionales de caucho.	*Versión estándar:* Lenguaje FORTH almacenado en la memoria ROM de 8 Kbytes.
PANTALLA	
Versión estándar: Dispone de una salida directamente conectable a un receptor en blanco y negro. *Formato de presentación:* 24 líneas de 32 caracteres. *Capacidad gráfica:* Resolución de 64 × 46 puntos.	

JUPITER ACE

dos casos se hace preciso desconectar y volver a conectar la clavija de la unidad de alimentación, pues el equipo carece de función RESET y de interruptor de red.

La pantalla

El ordenador doméstico que nos ocupa, fabricado por Jupiter Cantab Ltd., está preparado para conectarse directamente a un receptor de televisión, que puede ser perfectamente de blanco y negro, dado que el Jupiter ACE no trabaja con otros colores. La información se visualiza en formato de 24 líneas de 32 caracteres, generados por una matriz de 8 por 8 puntos que, como casi todo en este equipo, pueden ser modificados por el programador. El equipo funciona, tanto en modalidad texto como en modalidad gráfica. Para esta última dispone de un juego de caracteres gráficos y una serie de instrucciones entre las que destaca PLOT, capaz de hacer aparecer un cuadrado blanco o negro en cualquier punto de la pantalla. La resolución es de 64 por 46 puntos, quedando las dos últimas líneas reservadas al buffer de entrada. Además dispone de algunos comandos para tratamiento de gráficos, como DRAW (dibuja líneas rectas, uniendo pixels) INVIS y VIS (que evitan «ensuciar» el gráfico en caso de error). Las particulares características del lenguaje Forth permiten definir nuevas palabras que, una vez ejecutadas, se traducen en otras funciones gráficas.

Memorias de masa

Este ordenador dispone únicamente de la opción casete para el almacenamiento de programas y datos en memoria de masa. Para ello, precisa de un aparato de casete audio, preferiblemente no estereofónico, ya que los aparatos de mucha calidad tienden a alterar la señal haciéndola incomprensible para el ordenador.

La velocidad de transferencia entre el ordenador y la unidad de casete es de 1.500 baudios.

Los programas y datos pueden guardarse en el formato compacto del diccionario, así como verificarse, cargarse y mezclarse. Por otra parte, los bloques de memoria pueden quedar almacenados, verificados, cargados y reposicionados. Estas utilidades permiten incrementar la fiabilidad, por lo general no muy alta, de este sistema de almacenamiento.



El Júpiter Ace es el único ordenador doméstico que trabaja solamente con el lenguaje de programación FORTH. Esto le caracteriza como un sistema indudablemente original y de interés para los aficionados a la programación.

Periféricos

El Jupiter ACE admite adicionalmente un interface por medio del cual se puede conectar cualquier tipo de impresora matricial de alta velocidad, o de margarita para impresión de calidad, siempre que dichos periféricos dispongan del estándar de comunicación paralelo, tipo Centronics.

Asimismo, el Júpiter ACE es capaz de controlar automatismos exteriores por medio del interface SD1 (port de 6 bits de entrada-salida) que el fabricante suministra opcionalmente. Entre estos dispositivos, se pueden incluir sensores de humedad, detectores de incendios, alarmas antirrobo, termostatos, interruptores diversos, emisores, sirenas y marcadores telefónicos.

Por si lo anterior no resultara suficiente, la disposición de la conexión al bus del Júpiter ACE es similar a la del modelo ZX 81 de Sinclair Research. Esto se traduce en la posibilidad de utilizar periféricos concebidos para dicha máquina, siempre que dispongan del conector adecuado, en totales condiciones de compatibilidad.

Software

Las prestaciones del Júpiter ACE se deben a la utilización del Forth, una de cuyas ventajas es que permite crear extensiones partiendo de los propios comandos, sin necesidad de escribir rutinas en lenguaje máquina. El Forth resulta, indudablemente, más complejo que el BASIC. Sin embargo, cuenta con características que lo hacen especialmente interesante y que explican el número creciente de partidarios a ultranza de este lenguaje.

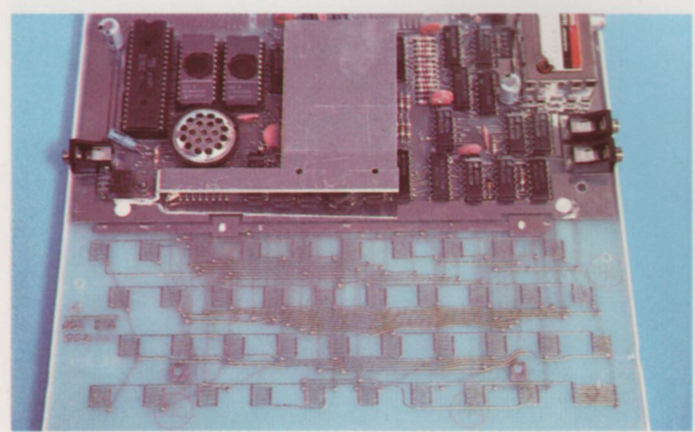
La misma tarea ocupa en Forth la mitad de memoria que en BASIC, y la ejecución de los programas es hasta diez veces más rápida. El Júpiter ACE puede compilar las palabras, cambiar palabras ya compiladas en el diccionario y ajustar todas las direcciones de compilación y comunicación. Dispone además de facilidades para la realización de operaciones de cálculo aritmético en coma flotante.

En Forth, la memoria se organiza en pilas de datos, y los programas deben estructurarse de manera que cada vez que se direccione un dato, éste se encuentre en el nivel superior de la pila.

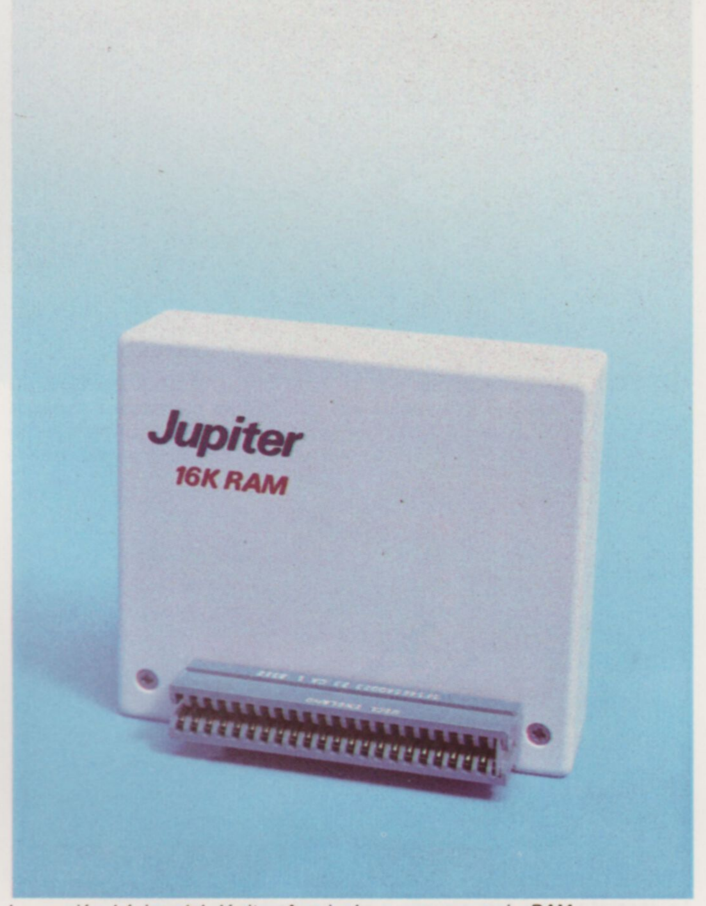
Esta máquina tiene, a disposición del usuario, un editor de programas capaz de listar y modificar cualquier palabra FORTH que hubiera sido definida y compilada previamente en el dicciona-



Esta máquina fue diseñada por un equipo que trabajó, antes de constituirse en empresa separada, con Clive Sinclair en el desarrollo del ZX-81, microordenador con el que guarda muchas similitudes, incluso, en su aspecto exterior.



La unidad central de este sistema se constituye alrededor de un Z-80 A, que trabaja a una frecuencia de 4 Mhz. La memoria ROM, de 8 Kbytes, contiene el lenguaje FORTH y todos los comandos de control del sistema.



La versión básica del Júpiter Ace incluye una memoria RAM de 3 Kbytes, que puede ser ampliada hasta 51 Kbytes mediante un módulo de extensión externo como el de la figura.

JUPITER ACE

rio, ya que el Júpiter ACE no utiliza pantalla de código fuente y toda entrada o salida de información queda registrada en un diccionario compilado, que se puede guardar en casete.

Entre las estructuras de control que guarda el sistema están IF-THEN-ELSE, DO-LOOP, BEGIN-WHILE-REPEAT, BEGIN-UNTIL. Todas ellas pueden ser mezcladas y conjuntadas a cualquier nivel.

Dado que el Forth es un lenguaje estructurado, la construcción o modificación de programas complejos resulta relativamente fácil. Un sistema de chequeo elimina el riesgo de destrucción accidental de los programas.

Software de aplicación

La biblioteca de programas de aplicación del Júpiter ACE es bastante amplia aun cuando se trata de un microordenador concebido para que sea el propio usuario quien desarrolle los programas adecuados a sus necesidades. El manual de programación que acompaña al equipo incluye diversos listados para el aprendizaje. El catálogo de programas para la versión de 3 Kbytes consta de un curso de programación en Forth, un juego de invasores, utilidades de programación, un monitor de código máquina, un calendario, biorritmos, conversiones, diferentes juegos de caracteres, y otras utilidades y juegos.

Entre los programas desarrollados para la versión de 19 Kbytes destacan un tratamiento de ficheros, un ensamblador, un generador de gráficos, Visicalc, funciones matemáticas, música, y un amplísimo surtido de programas de juegos.

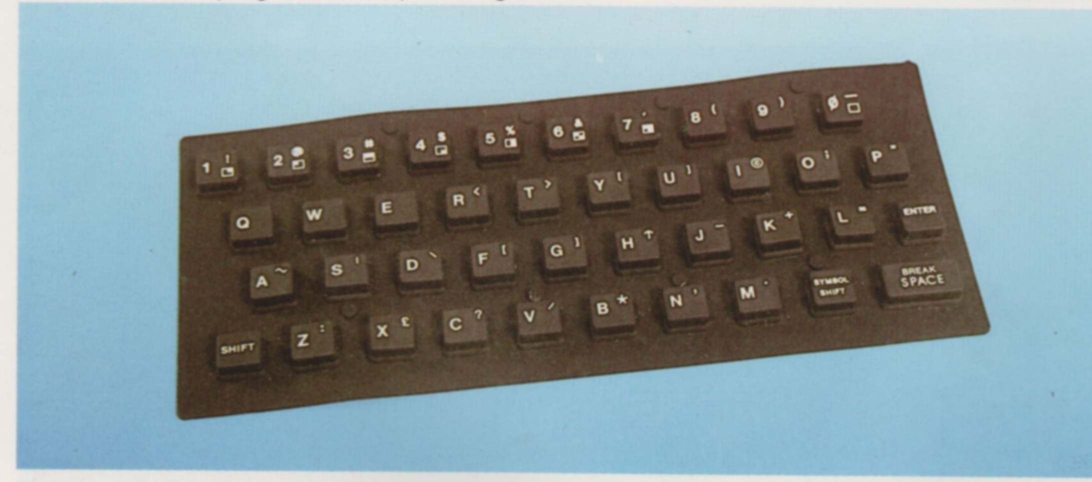
Soporte y distribución

Junto a la unidad central, se acompaña la fuente de alimentación y un manual de programación, además de un casete de demostración que contiene cinco archivos tipo diccionario, dos de los cuales sirven para introducir al usuario en el manejo de la máquina, y los otros tres son de juegos.

El Júpiter ACE se comercializa en España a través del importador Sushiro Data.

Configuración básica: Unidad central, que incluye el teclado de tipo QWERTY, con 3 Kbytes de memoria RAM y 8 Kbytes de ROM.

Configuración máxima: Unidad central con ampliación de memoria hasta un total de 51 Kbytes en RAM, monitor de televisión en blanco y negro, impresora paralelo, y unidad de casete para el almacenamiento masivo externo.



El teclado del Júpiter Ace consta de 40 teclas flotantes de caucho. A pesar de tener un aspecto muy sencillo, incluye símbolos particulares de programación como corchetes y llaves, y es, además, capaz de generar mayúsculas y minúsculas.



La conexión entre este sistema y sus periféricos se realiza a través de un bus externo al que se accede por la parte posterior de la carcasa. La mayoría de los dispositivos del ZX-81 son compatibles con él.



Con el microordenador se entrega un manual de programación que incluye varios listados para el aprendizaje del FORTH. La biblioteca de programas específicos para este sistema es bastante amplia.



PROGRAMACION MODULAR

SE puede decir que la programación modular es el sentido común aplicado a la resolución de un problema informático. Su objetivo es el programar en base a módulos, donde cada uno de éstos es una sección de programa que resuelve una parte del problema, de forma clara y determinada.

Cuando en un centro de cálculo, o a un nivel menor, en nuestro ordenador, se necesita crear «una aplicación», el primer objetivo es definir el resultado final que deseamos obtener.

Modificaciones en un programa no modular

Con el objetivo final en mente ya se puede empezar la codificación del programa (en el lenguaje que se hubiera considerado más necesario). Después de codificarlo y de probar con muchos datos que no tienen errores y que gestiona muy bien los periféricos, puede suceder que la Administración decide cambiar los porcentajes de retención, los intervalos de ciertas tablas, etc., que pueden ser los datos de partida del programa. Tristemente, el analista se da cuenta de que su trabajo anterior ya no le sirve: tiene entonces que modificarlo. Si no ha tomado precauciones, lo más probable es que no localice la parte afectada por estos cambios. Cuando la localiza, comprueba que un cambio de estas líneas del programa altera resultados empleados más adelante.

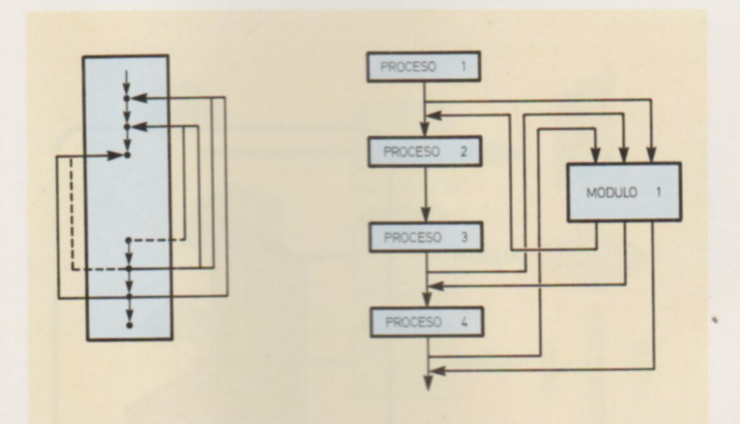
Si cada cosa estuviera en un sitio concreto, ya tendría mucho avanzado cuando tuviera que efectuar algún cambio. Si, además, todas las partes tuvieran una función muy delimitada, la modificación resultaría relativamente sencilla. Sabría dónde, cómo y por qué efectuar los cambios necesarios. Estaría haciendo programación modular.

Objetivos de un programa modular

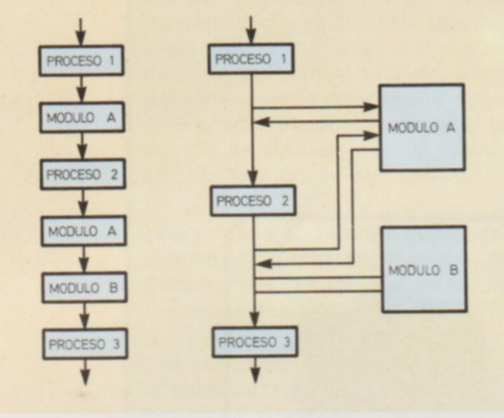
Los objetivos buscados con la programación modular son dos:

- Flexibilidad en los programas.
- Generalidad.

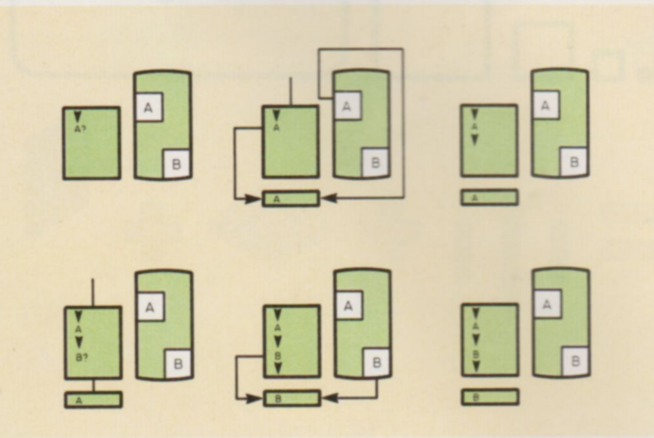
Flexibilidad: Cuando nuestros programas no son lo suficientemente «adaptables», y aparece un cambio exterior que afecta a sus datos de partida, lo normal es que se vuelvan a hacer o que no sirvan de nada. Debe ser posible, sin embargo, que el programa no sufra



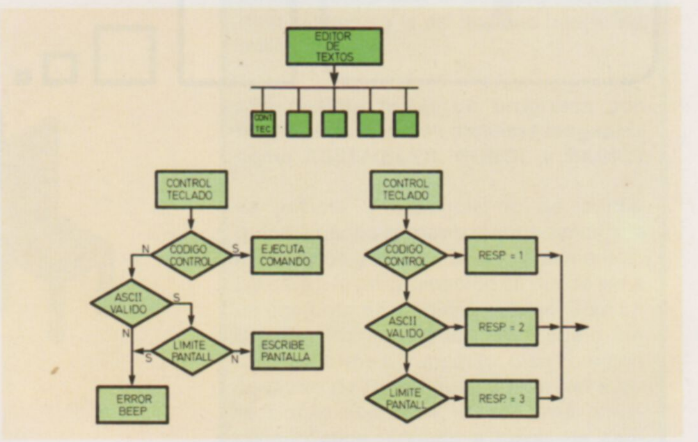
Ejemplo de organigrama de un programa no modular (a la izquierda), y de otro modular (a la derecha). En los programas no modulares las sentencias de salto pueden efectuarse desde cualquier parte del programa e ir a cualquier lugar de éste.



Un programa modular puede verse de dos formas diferentes: como una sucesión encadenada de módulos que se ejecutan uno tras otro, o como un conjunto de llamadas a subrutinas cerradas, y contenidas en cada uno de los módulos.



Cuando un programa llama a un módulo no almacenado en la memoria principal, el sistema lo trae del disco, lo graba en las posiciones correspondientes y le cede el control de la CPU.



Ejemplo de un programa de control de teclado realizado modularmente. Este programa es un módulo utilizado en una aplicación cualquiera de procesamiento de textos.

PROGRAMACION MODULAR

cambios profundos en la estructura y, muchísimo menos, en los datos internos: en este sentido son importantes los gestores de bases de datos. Una ley muy conocida de la informática, y a tener siempre en cuenta, dice que todo sistema cambia con el tiempo y que nunca se llega a delimitar hasta que se pretende mecanizar.

Generalidad: Un programa diseñado para cumplir con una serie de objetivos muy abiertos nunca usa la totalidad de las funciones previstas para ejecutar una aplicación concreta. El usuario está pagando entonces por algo que no está usando.

Puede llegar un momento en que ambos términos, flexibilidad y generalidad, se solapen y proporcionen así un resultado mucho más «interesante», tanto para el usuario como para el programador.

Programas transportables

Aparece entonces un nuevo concepto el de *transportabilidad*. Se puede considerar a esta característica como algo supra-programático, y es una de las máximas de todo departamento de análisis informático. Se aplica a cualquier programa que puede ejecutarse en un entorno para el que no fue creado ni pensado.

Se pueden considerar dos formas de transportabilidad:

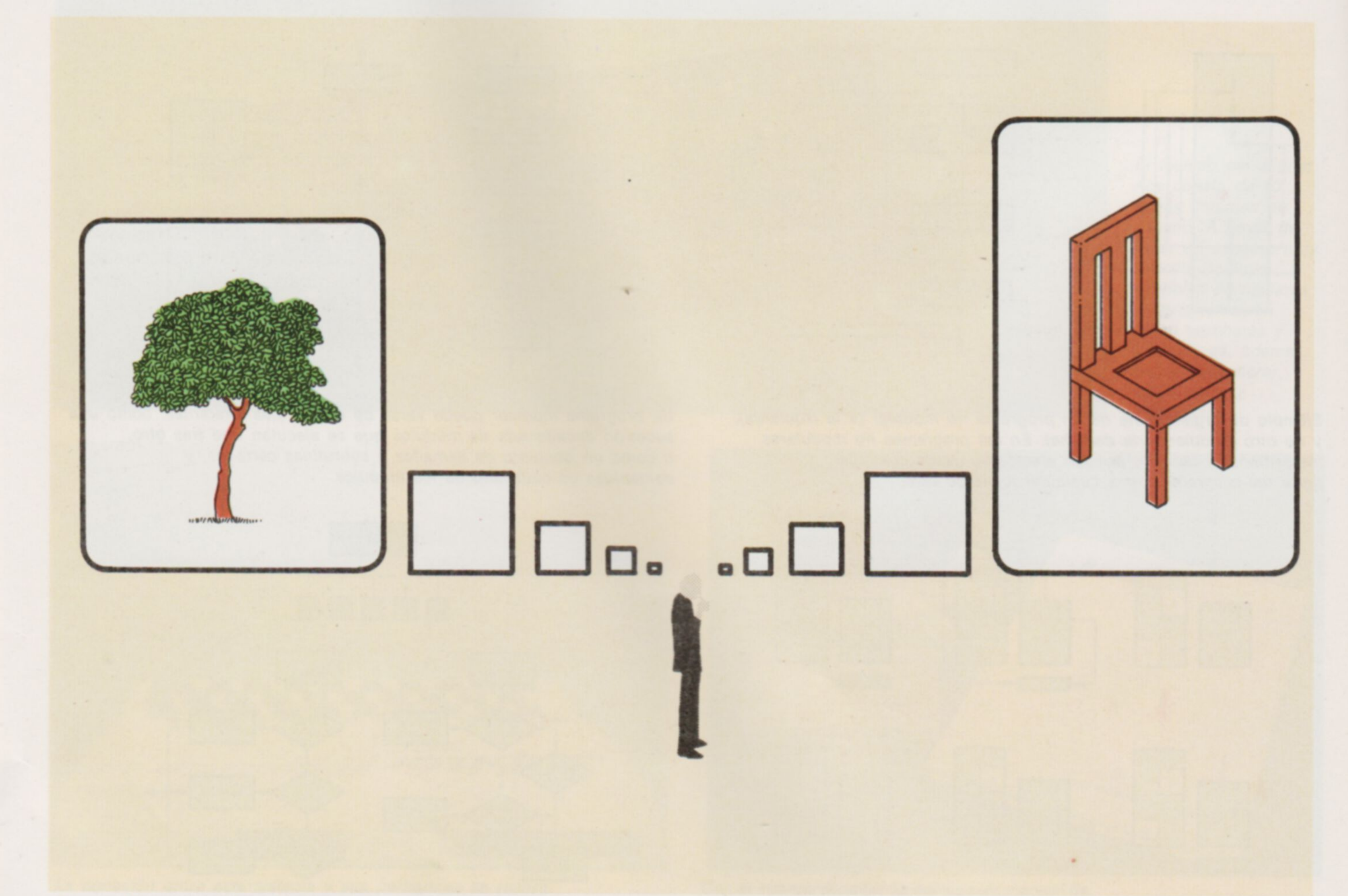
- Transportabilidad entre máquinas.
- Transportabilidad entre programas.

La transportabilidad entre máquinas se da cuando un programa que funciona en un sistema determinado se ejecuta en otro bastante distinto. Esta es una característica para la que se dan menos soluciones en el momento del diseño.

Mucho más normal es prever el intercambio entre aplicaciones dentro del mismo ámbito de la empresa o de la organización de una macroaplicación. Por ejemplo, si un programa necesita conocer todos los individuos que cumplen una condición determinada, se creará un «módulo» que haga «eso y sólo eso». Cuando para otra aplicación se necesite conocer los nombres de estas personas no se necesitará volver a escribir un programa: se intercalará directamente este módulo en la parte correspondiente de la aplicación.

Características de los módulos

La razón última de que un programa sea modular es que al subdividir un problema en varias tareas más pequeñas, directas y sencillas, es más fácil de escribir, leer y mantener.



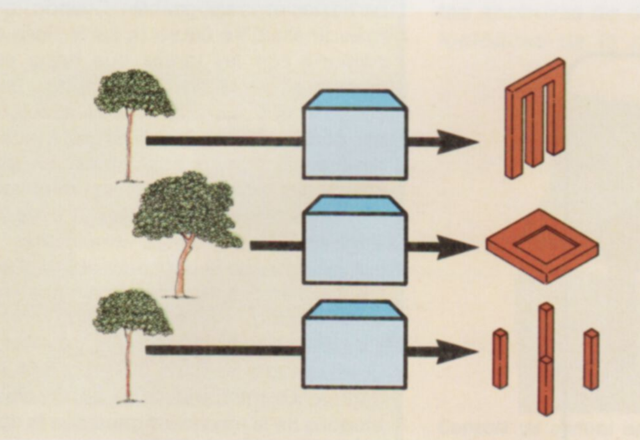
Los problemas no informáticos se pueden resolver también de forma modular. Un ebanista con sentido común, por ejemplo, fabrica sus muebles de acuerdo a reglas generales.

Cada módulo debe ser fácilmente visible e identificable. Cualquier otro programador debe ser capaz de encontrar, entender y modificar esta parte para que cumpla con nuevas especificaciones.

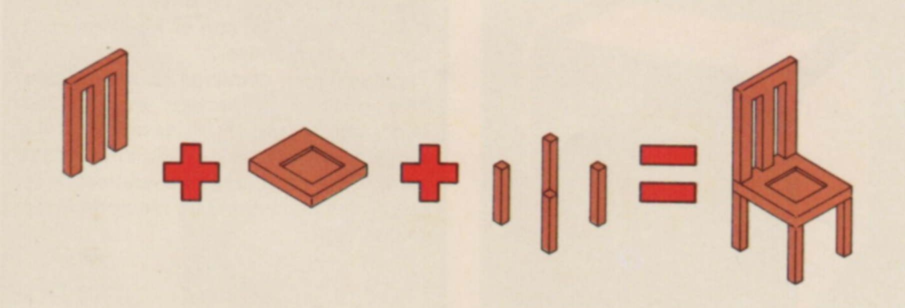
Un módulo debe ser independiente e integrarse sin ajustes a la estructura más o menos jerárquica del programa principal.

Podemos decir entonces que los módulos son partes que tienen una lógica propia, que son usados por un programa principal y que poseen una significación por sí mismos. La entrada y la salida deben estar bien definidas y ser únicas. Los módulos se asemejan a las ramas de un árbol, independientes entre sí, pero que a la vez están todas ellas unidas y relacionadas por el tronco. Este elemento director debe ser la serie de algoritmos de descomposición del problema en los módulos requeridos.

El algoritmo debe estar realizado para que sea fácil programar modularmente, y, además, siguiendo los criterios de *transportabilidad* y *flexibilidad* del conjunto. La modularidad está determinada por la elaboración de los algoritmos y por los análisis previos al diseño del sistema. El grado de modularidad es función de los conocimientos de informática y de la experiencia del programador ante estos problemas. Por ejemplo, la organización de un sistema operativo o de un compilador será tanto más flexible cuanto menor sea el número de parámetros fijos. Casi todos estos parámetros son contenidos de direcciones, donde estos contenidos son función de la configuración existente en ese momento.



Una silla se compone de cuatro patas iguales, del asiento, del respaldo, etc. Cada una de estas partes son los «módulos» de la silla.



Si se unen cada uno de los módulos, de la forma y en el orden adecuado, se obtiene el mueble deseado.

Glosario

¿Cómo se ejecutan varios programas que estén residentes en memoria al mismo tiempo?

En los sistemas pequeños el control de las interrupciones del sistema, permite saltar de un programa a la posición donde está otro de nuestros programas. En equipos mayores, el sistema operativo se encarga de la tarea de distribución del tiempo y del espacio para cada uno de los posibles módulos que en un momento dado residen en la memoria.

¿Puede el software de una máquina ejecutarse en otras diferentes sin problemas?

La compatibilidad del software es función de los compiladores o de los intérpretes de los sistemas utilizados. En equipos pequeños esta compatibilidad no suele darse, ya que cada uno de ellos suele usar muchas funciones específicas de la máquina que se trate, porque la gestión de discos varía de una máquina a otra, y, además, porque las características del procesador y las especificaciones de base no suelen coincidir.

¿Cómo sabe el ordenador cuándo ejecutamos un programa modular, dónde buscar y poner cada uno de los módulos especificados?

Cuando se compila cualquier programa la máquina produce un código objeto que no se carga directamente en la memoria de la máquina. El montador pasa las referencias externas del programa a la biblioteca de programas y utilidades a código objeto con la dirección del módulo llamador y la del llamado dentro del mismo ámbito.

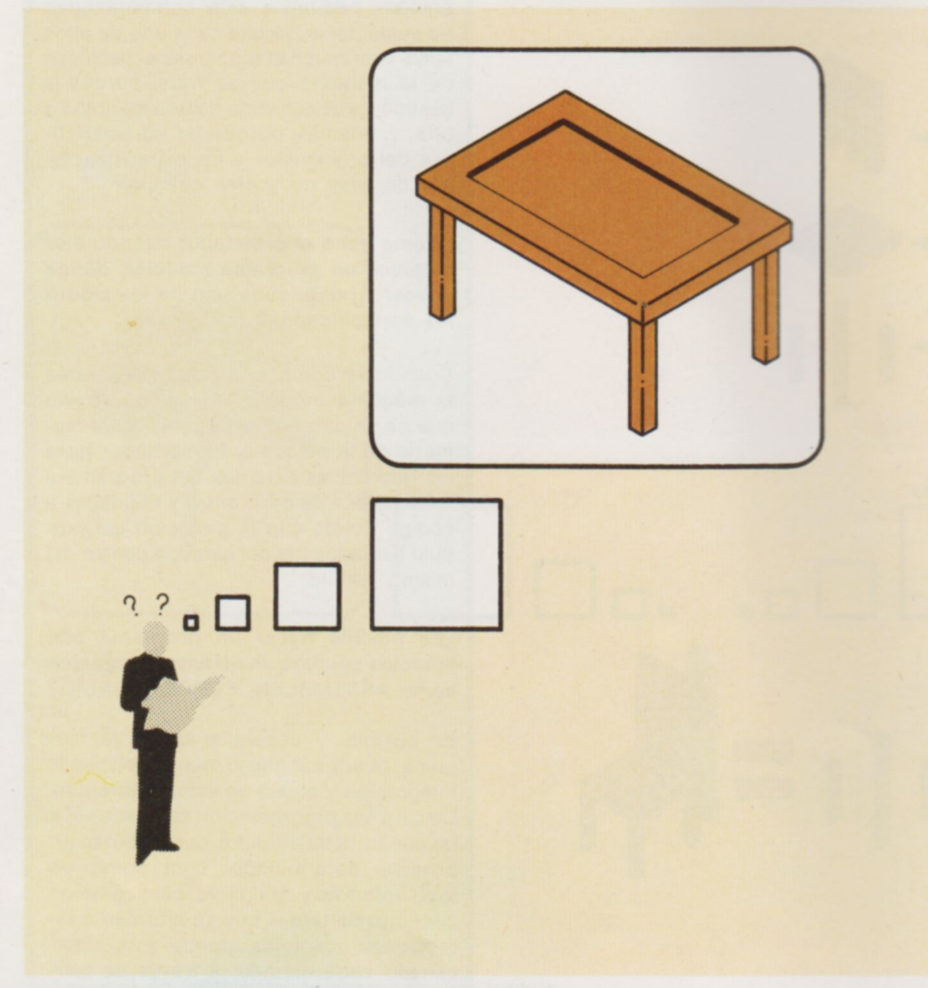
¿Es posible hacer un programa con módulos escritos en distintos lenguajes como ASSEMBLER, COBOL y BASIC?

Es posible, y dependiendo de la máquina, puede ser más o menos sencilla la integración y el uso de estos elementos. De cara a la programación se puede ver a un conjunto de módulos que ejecutan un proceso determinado, bien como un todo uniforme y compacto, bien como un conjunto de tareas que se bifurcan a direcciones absolutas, donde está almacenado cada módulo, a través de sentencias de carga de programa objeto.

PROGRAMACION MODULAR

Condiciones de modularidad

Para que se pueda hacer programación modular es necesario que exista un soporte tanto de hardware como de software adecuado. El hardware es preciso porque es impensable programar modularmente sin soportes de acceso directo y controladores independientes del procesador central. La responsabilidad recae, en este caso, sobre el software de base: sistema operativo, montador de enlace, reubicadores, etc.
El lenguaje, por su parte, debe contar con sentencias que permitan la existencia de este tipo de estructuras modulares en el programa. La sentencia más simple para escribir programas modulares es: DEF Función. Esta instrucción varía de unos sistemas a otros: normalmente ocupa sólo línea de programa, en la que se escriben los parámetros que debe tratar y devuelve un solo valor. En otros equipos, para ejecutar una función concreta, se salta a un punto del programa, usando variables propias de esta función. Esta operación puede ocupar, por supuesto, las líneas de programas necesarias. El retorno al programa principal puede hacerse desde cualquier punto del módulo, devolviendo o no variables.
Instrucciones no tan potentes o poco flexibles como DEF Función son, por ejemplo: GOSUB, CALL, USR, SYS, etc. Lo ideal es que la máquina trabaje con un lenguaje dotado de estructuras que permitan la implantación sencilla de módulos y de su interacción. Estos lenguajes son los llamados *estructurados*. Su principal característica es que permiten una gran variedad de maneras de segmentar, modular o encadenar a otros módulos existentes.



Cuando el ebanista desea fabricar una mesa puede recurrir a algunos de los módulos creados para la silla: las patas, por ejemplo.

Conceptos básicos

Asignación de memoria

La administración de la memoria es una de las funciones más importantes de un sistema operativo: para aumentar el rendimiento de la ejecución múltiple de programas es necesario un control eficaz y flexible del espacio de memoria direccionable.
El sistema operativo de un ordenador de tiempo compartido debe controlar que ninguna tarea entre en las zonas de memoria de los demás o del propio sistema. A pesar de todo, pueden existir zonas de memoria contiguas. No se trae a ejecutar, de esta forma, ningún otro programa por falta de espacio suficiente.
Una solución es la partición reasignable de memoria o la paginación mediante la participación reasignable. Cuando una tarea finaliza, se cubre la parte libre de memoria con las tareas que hasta ese momento se están procesando. Esto obliga a que las direcciones absolutas del código objeto cambien para adaptarse a las nuevas posiciones. En este caso todas las direcciones son relativas a una determinada, la del registro de reasignación. Las nuevas direcciones se hallan sumando la dirección última a una cantidad de desplazamiento almacenado en el registro de reasignación.
La asignación paginada crea un espacio direccionable de cierta longitud, llamado página. La memoria física se divide en bloques de la misma longitud que la de la página. Cualquier página puede almacenarse, por tanto, en cualquier bloque. Cada uno de estos bloques tiene un índice para el calculo de la dirección física. Cualquier tarea se divide en un número de páginas, a cada una de las cuales se asigna un bloque. La dirección se calcula, en este caso, con el índice de cada uno de los bloques.
Paginación por demanda es otro método de asignación de memoria, que también se conoce por el nombre de memoria virtual. En este caso el espacio direccionable por los programas generalmente es mayor que la cantidad de memoria física disponible.

PLOTTERS BENSON

La firma BENSON comercializa una variada y amplia gama de plotters, tanto de pluma como electrostáticos. Dentro de los plotters de pluma esta firma dispone de plotters de mesa (flat-bed), de rodillo y de tambor.

Plotters de mesa

Dentro de este tipo existen distintos modelos: 1422, 1423, 1425, 1455. Las características de estos modelos están reflejadas en la tabla correspondiente. La sujección del papel a la mesa se efectúa de forma electrostática.
El dibujo puede ser realizado mediante tres tipos de plumas: rotuladores de bola, bolígrafos de tinta presurizada o plumas de tinta china. La velocidad de dibujo varía en cada caso. En el modelo 1425, por ejemplo, la velocidad axial de las plumas, trabajando con rotulador, es de 50 cm/seg; de 50 ó 70 cm/sg, utilizando bolígrafos de tinta presurizada, y de 40 cm/seg, usando plumas de tinta china.

Estos plotters tienen los controles y la electrónica en una unidad separada, dotada de una pantalla con teclado, mediante el cual el operador puede introducir y modificar los parámetros que desee.

Una característica diferenciadora del modelo 1425 es la posibilidad de efectuar rotaciones del dibujo de 90°, 180° ó 270°.

Todos los modelos disponen de interface del tipo paralelo y del tipo serie RS232.

Plotters de rodillo

De este tipo existen varios modelos. Cada uno de ellos proporciona distintas anchuras de dibujo y diferentes velocidades de la pluma.

Dentro de estos modelos, el más alto de la gama es el 1342. Este plotter posee cuatro velocidades programables de la pluma: 20, 30, 65 y 80 cm/seg. Las aceleraciones correspondientes a estas velocidades son: 3 g, 6 g, 3 g y 6 g. Con este plotter se puede conseguir una máxima velocidad diagonal de la pluma de 113 cm/seg, con una alteración de 8,5 g.
Al igual que en los plotters de mesa, el dibujo puede ser realizado mediante rotuladores de bola, bolígrafos de tinta presurizada o plumas de tinta china.

Plotters de tambor

El modelo 1565 utiliza un tambor ultraligero de 40 cm de diámetro, indeformable, que gira en ambos sentidos. Lleva un cabezal portaútiles que se mueve paralelamente al eje del tambor. Este plotter puede utilizar papel de cualquier tipo y tamaño, de hasta



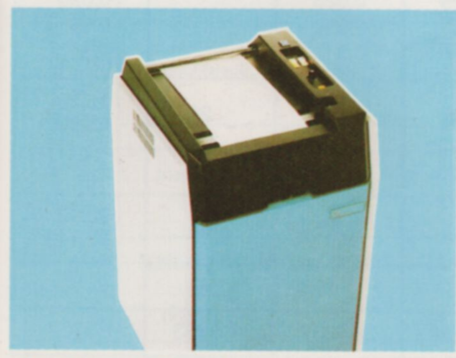
Plotter de mesa Benson 1425. El papel se sujeta a la mesa de forma electrostática. Este modelo puede efectuar rotaciones del dibujo de 90, 180, ó 270 grados.



Consola de control del plotter 1425. La electrónica del periférico está alojada en su interior. Mediante el teclado y la pantalla pueden modificarse los parámetros del dibujo.



Plotters de rodillo. Ejecutan el dibujo mediante rotuladores de bola, bolígrafos de tinta presurizada o plumas de tinta china.



Los plotters electrostáticos proporcionan una alta velocidad de dibujo. Los modelos Benson de este tipo pueden utilizarse, además, como impresoras, ya que son capaces de generar caracteres alfabéticos.



La mayor anchura de papel con la que trabajan los plotters electrostáticos de esta firma es de 44 pulgadas. El de la figura tiene una resolución de dibujo de 254 puntos por pulgada.



Todos estos plotters se suministran con su software correspondiente, que incluye el conjunto de subrutinas gráficas de Benson, escritas en Fortran o en Basic.

PLOTTERS BENSON

1.200 × 864 mm, que se sujeta al tambor mediante zonas autoadhesivas.
Las características más destacables de este modelo son:

- Tamaño de dibujo: 1.200 × 864 mm.
- Velocidad de la pluma: 80 cm/seg.
- Velocidad de la pluma con la pluma levantada: seleccionable, entre 80 cm/seg y 113 cm/seg.
- Aceleración: 4 g ó 5,6 g.
- Número de plumas: 4.
- Precisión: 0,1 mm.
- Resolución: Programable entre 0,0125, 0,025, 0,05 y 0,1 mm.
- Repetibilidad: ± 0,05 mm.

Plotters electrostáticos

En los plotters electrostáticos el movimiento del papel es unidireccional. Este se carga eléctricamente al pasar frente a un peine de electrodos, después de lo cual es entintado y secado.
La principal característica de este tipo de plotters es su rapidez de dibujo, ya que el tiempo máximo de ejecución de un plano tamaño DIN A0 es de dos minutos, independientemente de la complejidad del plano.

Estos plotters , gracias a su posibilidad de generar caracteres, pueden utilizarse también como impresoras de alta velocidad. El modelo 9636 tiene una anchura de papel seleccionable entre 28, 56, 61, 92 y 112 cm, una resolución de 900 puntos por pulgada y una velocidad de avance del papel de hasta 5 cm/seg.

Firmware

Todos los plotters de la firma BENSON tienen dos niveles de software, almacenados en la memoria interna del microprocesador. Estos dos niveles se llaman niveles de inteligencia I0 e I1. Mediante este firmware se pueden generar círculos, arcos, distintos tipos de trazos, caracteres, etc.
Las posibilidades más interesantes del nivel I0 son las siguientes:

- Generación de círculos.
- Generación de 96 caracteres y 96 símbolos.
- Orientación de textos.
- Buffer de 2 Kbytes.
- Interface paralelo e interface RS232.

Las posibilidades más importantes. El nivel I1 destaca por las siguientes características:

- Generación de círculos y arcos.
- 16 tipos de trazos.
- Generación de 256 caracteres y 128 símbolos.
- Posibilidades de rotación, escala, inclinación, espaciado variable.
- Generación de subíndices y superíndices.
- Orientación de textos, centrado y reparto proporcional.
- Buffer de 2 Kbytes.
- Interface paralelo, RS232 e IEEE 488.

Software

Existen dos tipos de rutinas de software estándar, con las cuales puede dialogar cualquiera de los plotters BENSON:

- ISDP.
- GPR 100.

ISPD es una rutina escrita en su mayor parte en FORTRAN IV y está disponible en la mayoría de ordenadores.
GPR 100 es un paquete de rutinas escritas en FORTRAN, que permite una utilización sencilla de las posibilidades de la inteligencia gráfica de los plotters de pluma y electrostáticos.

CARACTERISTICAS DE PLOTTERS DE RODILLO

		1122	1132	1222	1232	1322	1332	1333	1342
Anchura de dibujo (cm)		32	32	73	73	93	93	93	93
Máxima longitud de dibujo (m)		50	50	50	50	50	50	50	50
Número de plumas		3	3	4	4	4	4	4	4
Resolución (mm)		0,0125 0,025 0,05 0,1	0,0125 0,025 0,05 0,1	0,0125 0,025 0,05 0,1	0,0125 0,025 0,05 0,1	0,0125 0,025 0,05 0,1	0,0125 0,025 0,05 0,1	0,025	0,0125 0,025 0,05 0,1
Velocidad axial de la pluma (cm/seg)	Pluma abajo	15	25	15	25	15	25	25	20,30 65,80
	Pluma arriba	15	31	15	31	15	31	35	
Velocidad diagonal de la pluma (cm/seg)	Pluma abajo	21	35	21	35	21	35	35	28,42 92,113
	Pluma arriba	21	45	21	45	21	45	45	
Tipo de interface		RS232	RS232	RS232	RS232	RS232	RS232	RS232	RS232
Temperatura de funcionamiento		18°C÷30°C	18°C÷30°C	18°C÷30°C	18°C÷30°C	18°C÷30°C	18°C÷30°C	18°C÷30°C	18°C÷30°C
Humedad de funcionamiento		25%÷80%	25%÷80%	25%÷80%	25%÷80%	25%÷80%	25%÷80%	25%÷80%	25%÷80%
Consumo (V. A.)		500	500	500	500	500	500	500	1.500

CARACTERISTICAS DE PLOTTERS DE MESA				
	1422	**1423**	**1425**	**1455**
Tamaño de dibujo (cm)	A0 84 × 120	A0 84 × 120	A0 84 × 120	A0 120 × 168
Número de plumas	4	4	4	4
Velocidad de la pluma (cm/seg)	30	30	50	50
Aceleración	0,5 g	0,5 g	1,5 g	1,5 g
Resolución (mm)	Programable: 0,0125 0,025 0,05 0,1	Programable: 0,0125 0,025 0,05 0,1	Programable: 0,0125 0,025 0,05 0,1	Programable: 0,0125 0,025 0,05 0,1
Precisión (mm)	0,1	0,1	0,1	0,1
Repetibilidad (mm)	± 0,025	± 0,025	± 0,025	± 0,025

CARACTERISTICAS DE PLOTTERS ELECTROSTATICOS								
			9322	**9336**	**9344**	**9424**	**9436**	**9444**
Anchura de papel			22″	36″	44″	24″	36″	44″
Anchura de dibujo (cm)			53,64	89,41	109,73	59,51	89,59	108,79
Velocidad de dibujo		i.p.s.	1,5	1	0,6	1	0,5	0,35
		m²/min	1,28	1,39	1,02	0,93	0,7	0,6
Resolución (puntos/pulgada)			200	200	200	254	254	254
FUNCIONAMIENTO COMO IMPRESORA	Juego de caracteres		123	123	123	123	123	123
	Caracteres por pulgada	Tamaño pequeño	12,5	12,5	12,5	15,8	15,8	15,8
		Tamaño normal	6,25	6,25	6,25	7,9	7,9	7,9
	Caracteres por línea	Tamaño pequeño	264	440	540	372	560	680
		Tamaño normal	132	220	270	186	280	340
	Líneas por pulgada	Tamaño pequeño	6	6	6	8	8	8
		Tamaño normal	3	3	3	4	4	4
	Velocidad de impresión (l.p.m.)	Tamaño pequeño	560	370	230	470	238	160
		Tamaño normal	280	185	115	235	119	80
Temperatura de funcionamiento			5°C÷40°C	5°C÷40°C	5°C÷40°C	5°C÷40°C	5°C÷40°C	5°C÷40°C
Humedad de funcionamiento			15%÷85%	15%÷85%	15%÷85%	15%÷85%	15%÷85%	15%÷85%
Consumo (V. A.)			550	550	550	550	550	550

CONTABILIDAD FACIT DTC

EL programa de Contabilidad General para FACIT DTC permite la mecanización de las operaciones contables, utilizando el Plan General Contable Español u otro definido por el usuario dentro de ciertos márgenes de coherencia. Posee una capacidad para 3.000 cuentas y 4.800 asientos dobles en la versión de 320 Kbytes, y 1.500 cuentas y 2.000 asientos dobles, en la versión de 160 Kbytes.

Cada cuenta viene definida por un número de 7 dígitos: 3 para la cuenta principal y 4 para la auxiliar. A cada cuenta se le asigna el número de bloque que corresponde con el número de orden que ocupa en el fichero. Al definir el Plan de Cuentas es conveniente dejar bloques vacíos con vistas a un futuro crecimiento.

Creación de ficheros

Mediante esta opción, se crean los ficheros MAYOR y DIARIO, en blanco, en sus discos correspondientes.

Inicialización

En esta opción se introducen datos relativos a la empresa, tales como su nombre, dirección, teléfono, etc., así como un código que servirá de identificación y para comprobar los discos de datos.

Incluye también la posibilidad de listar sólo cuentas abiertas en el Plan o todo el fichero; listar todo el Mayor o sólo las cuentas con movimientos, etc.

Definición de cuentas

Contiene los siguientes campos el fichero: cuenta principal, cuenta auxiliar, descripción, Debe y Haber. Las cantidades asignadas al Debe y Haber son las de apertura, y no se modifican hasta el cierre del ejercicio.

Listado del plan de cuentas

Contiene todos los datos mencionados, y el saldo de la cuenta si se desea, pudiendo obtenerse como un listado parcial.

Introducción de asientos

Se introducen los siguientes datos: mes, día, número de documento, número de bloque del Debe y número de

Aplicación: **Contabilidad general**
Aplicación: **FACIT DTC**
Configuración: **Unidad central, teclado, pantalla, doble unidad de discos de 5 25" e impresora**
Memoria requerida: **30 Kbytes**
Soporte: **Unidad de discos flexibles de 320 Kbytes**
Sistema operativo: **CP/M, ABC**
Documentación: **Manual de 15 páginas en español**
Copyright: **NOVOMATICA, S. A.**
Distribuidor: **NOVOMATICA, S. A.**

LISTADOS EDITABLES

Plan de cuentas.
Diario.
Mayor.
Ficha de cuenta.
Balance.
Cuenta de explotación.
Diario de cierre.
Diario de reapertura.



Esta aplicación contable se ejecuta sobre un ordenador Facit DTC. Los periféricos necesarios para su desarrollo son una unidad de disco flexible de 330 Kbytes y una impresora.

bloque del Haber, seguidos de descripción e importe. Si el asiento tiene partida/contrapartida simple, ocupa un solo registro. El programa va sumando el Debe y el Haber para verificar el cuadre y se pueden solicitar los acumulados pulsando «&».

Listado del diario

Efectúa el listado de los apuntes introducidos desde el principio del período, totalizando al final los importes para verificar si existe error de cuadre.

Corrección de asientos

Permite modificar los datos introducidos en un asiento, e incluso, anularlo.

FICHEROS Y CAPACIDADES

Fichero	Disco	Capacidad
Mayor	320 Kbytes	3000 Regs
Diario	320 »	4800 »
Mayor	160 »	1500 »
Diario	160 »	2000 »

```
1) CREACION DE FICHEROS
2) INICIALIZACION
3) DEFINICION DE CUENTAS DE MAYOR
4) LISTADO DE CUENTAS
5) INTRODUCCION DE ASIENTOS
6) LISTADO DE DIARIO
7) CORRECCION DE ASIENTOS
8) PREPARACION DE MAYOR
9) EJECUCION DE MAYOR
0) BALANCE/CUENTA EXPLOTACION
A) CONSULTA DE CUENTAS
B) CIERRE/APERTURA
C) FORMATEO DE DISCOS
D) COPIA DE DISCOS
*) TERMINAR
```

El menú general de la aplicación es único y consta de 15 puntos. Permite la creación, modificación, baja y consulta de los dos ficheros de datos, Diario y Mayor.

Preparación del Mayor

Este programa encadena las cuentas con sus asientos correspondientes.

Ejecución del Mayor

Lista del Mayor de las cuentas seleccionadas, según la opción elegida en la inicialización. Permite la ejecución sin tener que listar por impresora, para el caso de una modificación de Diario sin desear otro listado del Mayor completo.

Balance/Cuenta de explotación

Realiza el listado a nivel de detalle o general, es decir, citando cada una de las cuentas auxiliares. Los listados se efectúan en pantalla o en impresora.

Consulta de cuentas

Permite consultar el estado de una cuenta en particular por pantalla o impresora.

Cierre/Apertura

Genera los datos de cierre y apertura y pone los saldos de las cuentas de Mayor a cero. Para la apertura se debe tener un disco de Diario creado en blanco mediante las opciones 1 y 2. Para la apertura se debe utilizar el disco de Diario nuevo y el de Mayor, ya creado.

LISTADO DEL PLAN DE CUENTAS

NUM.BLOQ.	CUENTA PPAL.	AUX.	DESCRIPCION	SUMAS DEBE	HABER	SALDOS DEUDOR	ACREEDOR
0001	100	0000	CAPITAL SOCIAL	0,00	500.000,00		500.000,00
0002	110	0000	PRIMA EMISION ACCIONES	0,00	0,00	0,00	
0003	111	0000	PLUSVALIA REV. ACTIVO	0,00	0,00	0,00	
0004	113	0000	RESERVAS LEGALES	0,00	0,00	0,00	
0005	115	0001	FONDOS RESERVA INVERSION	0,00	0,00	0,00	
0006	115	0002	RESERVA ESPECIAL REINVERSION	0,00	0,00	0,00	
0007	120	0000	PREV. PARA RIESGOS	0,00	0,00	0,00	
0009	130	0000	REMANENTE	0,00	0,00	0,00	
0010	131	0000	RESULTADO NEGAT.EJERC. 1980	255.929,00	0,00	255.929,00	
0011	132	0000	RESULTADO NEGAT.EJERC. 1981	280.099,34	0,00	280.099,34	
0012	133	0000	RESULTADO NEGATIVO EJERCICIO	0,00	0,00	0,00	
0013	140	0000	SUBVENCIONES OFICIALES	0,00	0,00	0,00	
0014	141	0000	OTRAS SUBVENCIONES	0,00	0,00	0,00	
0015	170	0001	PREST.PLAZO LARGO BANCOBAO	0,00	0,00	0,00	
0020	175	0001	ACREEDORES A PLAZO LARGO	0,00	0,00	0,00	
0025	180	0000	DEPOSITOS A PLAZO LARGO	0,00	0,00	0,00	
0026	190	0000	ACC. CAPITAL SIN DESEMBOLSAR	0,00	0,00	0,00	
0027	191	0000	ACC. PRIMA EMIS. SIN DESEMB.	0,00	0,00	0,00	
0028	193	0000	ACC. PROPIOS SITUAC. ESPEC.	0,00	0,00	0,00	
0029	200	0000	TERRENOS Y BIENES NATURALES	0,00	0,00	0,00	
0030	202	0000	EDIFICIOS Y OTRAS CONSTRUCC.	0,00	0,00	0,00	
0031	203	0001	MAQUINARIA	0,00	0,00	0,00	
0032	203	0002	INSTALACIONES	129.561,00	0,00	129.561,00	
0033	203	0003	UTILES Y HERRAMIENTAS	53.253,00	0,00	53.253,00	
0034	204	0000	ELEMENTOS DE TRANSPORTE	0,00	0,00	0,00	
0035	205	0001	MOBILIARIO	144.654,00	0,00	144.654,00	
0036	205	0002	EQUIPOS DE OFICINA	0,00	0,00	0,00	
0037	205	0003	MATERIAL DE OFICINA	49.711,00	0,00	49.711,00	
0038	205	0004	EQUIPOS TECNICOS	0,00	0,00	0,00	
0039	206	0000	EQUIPOS PROCESO INFORMACION	0,00	0,00	0,00	
0040	211	0001	PATENTES	0,00	0,00	0,00	
0041	211	0002	MARCAS Y NOMBRES COMERC.	0,00	0,00	0,00	
0042	211	0003	LICENCIAS Y PROCED. FABRIC.	0,00	0,00	0,00	
0043	211	0004	MODELOS DE UTILIDAD	0,00	0,00	0,00	
0044	233	0000	MAQUINAR.E INST.EN MONTAJE	0,00	0,00	0,00	
0045	236	0000	EQUIP.PROC.INFORM. MONTAJE	0,00	0,00	0,00	
0046	239	0001	GASTOS ESTD.E INVEST.TECNIC.	900,00	0,00	900,00	
0047	239	0002	GASTOS ESTUDIOS COMERCIALES	0,00	0,00	0,00	
0048	260	0000	FIANZAS A PLAZO LARGO	177.800,00	77.800,00	100.000,00	
0049	265	0000	DEPOSITOS A PLAZO LARGO	0,00	0,00	0,00	
0050	270	0000	GASTOS DE CONSTITUCION	130.394,00	0,00	130.394,00	
0051	271	0000	GASTOS PRIMER ESTABLECIM.	0,00	0,00	0,00	
0052	272	0000	GASTOS AMPLIACION DE CAPITAL	0,00	0,00	0,00	
0053	273	0000	GASTOS PUESTA EN MARCHA	0,00	0,00	0,00	
0054	275	0000	GASTOS FORMALIZ. PRESTAMOS	0,00	0,00	0,00	
0055	277	0000	INVEST. ESTU. PROY. A AMORT.	0,00	0,00	0,00	
0056	278	0000	MODIF. PAS. MONET. TIPO CAMB	0,00	0,00	0,00	
0057	280	0001	DE EDIFIC. Y OTRAS CONSTRUC.	0,00	0,00	0,00	

Mediante la opción cuarta del Menú general se obtiene el listado impreso del Plan de cuentas, prácticamente equivalente al Balance.

BALANCE PAGINA 1

CUENTAS	SUMAS DEBE	HABER	SALDOS DEUDORES	ACREEDORES
100 . 0000 CAPITAL SOCIAL	0,00	500.000,00		500.000,00
TOTAL CTA. 100	0,00	500.000,00	0,00	500.000,00
TOTAL SUBGPO. 10	0,00	500.000,00	0,00	500.000,00
110 . 0000 PRIMA EMISION ACCIONES	0,00	0,00	0,00	
TOTAL CTA. 110	0,00	0,00	0,00	0,00
111 . 0000 PLUSVALIA REV. ACTIVO	0,00	0,00	0,00	
TOTAL CTA. 111	0,00	0,00	0,00	0,00
113 . 0000 RESERVAS LEGALES	0,00	0,00	0,00	
TOTAL CTA. 113	0,00	0,00	0,00	0,00
115 . 0001 FONDOS RESERVA INVERSION	0,00	0,00	0,00	
115 . 0002 RESERVA ESPECIAL REINVERSION	0,00	0,00	0,00	
TOTAL CTA. 115	0,00	0,00	0,00	0,00
TOTAL SUBGPO. 11	0,00	0,00	0,00	0,00
120 . 0000 PREV. PARA RIESGOS	0,00	0,00	0,00	
TOTAL CTA. 120	0,00	0,00	0,00	0,00
TOTAL SUBGPO. 12	0,00	0,00	0,00	0,00
130 . 0000 REMANENTE	0,00	0,00	0,00	
TOTAL CTA. 130	0,00	0,00	0,00	0,00
131 . 0000 RESULTADO NEGAT.EJERC. 1980	255.929,00	0,00	255.929,00	
TOTAL CTA. 131	255.929,00	0,00	255.929,00	0,00
132 . 0000 RESULTADO NEGAT.EJERC. 1981	280.099,34	0,00	280.099,34	
TOTAL CTA. 132	280.099,34	0,00	280.099,34	0,00
133 . 0000 RESULTADO NEGATIVO EJERCICIO	0,00	0,00	0,00	
TOTAL CTA. 133	0,00	0,00	0,00	0,00
TOTAL SUBGPO. 13	536.028,34	0,00	536.028,34	0,00
140 . 0000 SUBVENCIONES OFICIALES	0,00	0,00	0,00	
TOTAL CTA. 140	0,00	0,00	0,00	0,00

El listado del Balance se puede ejecutar por impresora o por pantalla. El usuario puede pedir los datos de esta cuenta de forma detallada o general.

PROGRAMA

Título: ***Raid***
Ordenador: ***Dragon 32***
Memoria requerida: ***32 Kbytes***
Lenguaje: ***BASIC***

La práctica totalidad de los microordenadores cuentan en su biblioteca de programas con un programa similar al que hoy presentamos.

El jugador tripula un bombardero que cada vez vuela más bajo y debe destruir todos los bloques de casas que se encuentran debajo de él para poder aterrizar. La altura de los edificios es aleatoria y el número de bombas de que dispone el piloto está limitado a tres por pasada. Para disparar la carga destructiva debe pulsar la barra espaciadora.

Cuando en el Dragon se pulsa una tecla el código de ésta pasa al Keyboard buffer (algo así como una memoria de la última tecla pulsada) y allí se almacena hasta que se envía un comando de utilización (INPUT, INKEYS, etc.), o bien hasta que se envía un comando de utilización (INPUT, INKEYS, etc.), o bien bas y se pulsa la tecla espaciadora por cuarta vez, la orden de lanzamiento queda almacenada en el buffer del teclado. De esta forma, cuando el bombardero se encuentre en el comienzo de la siguiente pasada se ejecutará un INKEYS y se dejará caer la bomba.

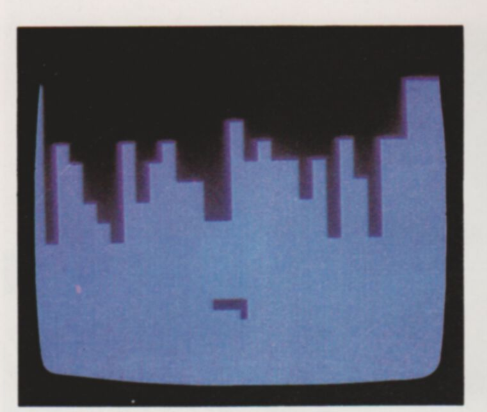
Este efecto, que puede parecer molesto, no ha sido eliminado del programa porque proporciona una gran ventaja: El primero de los edificios es siempre el más difícil de derribar y este «disparo automático» facilita esta labor enormemente. Por otro lado, existe una forma sencilla de anular este efecto que consiste en pulsar otra tecla cualquiera con lo que su código pasará a sustituir a la orden de lanzamiento. Al ser leída en la siguiente pasada, no será considerada como disparo.

Si se consigue aterrizar, el programa informará del número de bombas empleadas y si es, o no, el mejor récord obtenido hasta el momento.

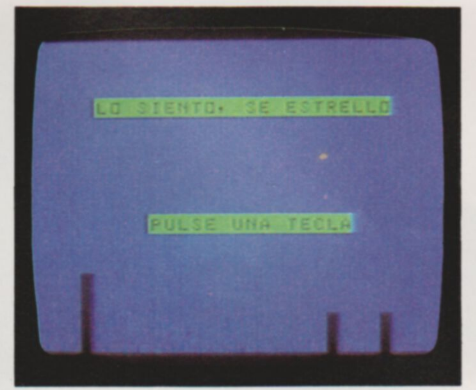
CUADRO DE VARIABLES

Variable	Descripción	Variable	Descripción
A	Bombas que quedan por pasada.	L	Frecuencia para el SOUND de la línea 290 dependiente de la altura del avión.
B	Número total de bombas utilizadas.	R	Récord.
I	Parámetro de posicionamiento del avión.	A ( )	Tabla que contiene la altura de los bloques.
J	Como la anterior.	AS	Parámetro de diversa utilidad.
K	De diversa utilidad.	BS	Bombardero.

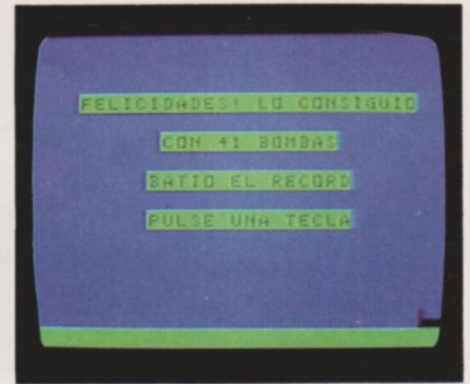
```
10 REM LOPEZ MARTINEZ * RAID * DRAGON 32
20 R=10E37
30 DIMA(32)
40 CLS3
50 B$=CHR$(175)+CHR$(164)+CHR$(172)+CHR$(173)
60 FORI=0TO2:A$=A$+CHR$(175):NEXT
70 B=0
80 FORI=0TO32:A(I)=-1:NEXT
90 FORI=484TO510
100 K=RND(8)-1
110 A(I-480)=K
120 FORJ=I TO I-32*K STEP-32
130 PRINT@J,CHR$(192);
140 NEXTJ,I
150 FORI=0TO480STEP32
160 A=3
170 FORJ=0TO28
180 IF15-A(J+3)=INT(I/32)THEN460
190 PRINT@I+J,B$;
200 IF A=0THEN340
210 IFINKEY$<>" " OR I>479 THEN340
220 B=B+1
230 A=A-1
240 K=I+J+2
250 L=10*(16-INT(I/32))
260 GOTO280
270 PRINT@K,CHR$(175);
280 PRINT@K+32,CHR$(167);
290 SOUNDL,1
300 IFK<448 THEN K=K+32:GOTO270
310 PRINT@K+32,CHR$(175);
320 A(J+2)=-1
330 GOTO350
340 FORK=0TO24:NEXT
350 NEXT
360 IFI=480THEN 380
370 PRINT@I+29,A$;
380 NEXT
390 PRINT@100,"FELICIDADES! LO CONSIGUIO";
400 PRINT@170,"CON";B;"BOMBAS";
410 IFB<R THEN R=B:PRINT@233,"BATIO EL RECORD";
420 PRINT@297,"PULSE UNA TECLA";
430 IFINKEY$<>""THENGOTO440
440 IF INKEY$=""THEN440
450 GOTO40
460 FORK=50 TO 1 STEP-10
470 SOUNDK,1
480 A$="":FORL=0TO2:A$=A$+CHR$(159+16*INT(K/10)):NEXT
490 PRINT@I+J,A$;
500 NEXT
510 PRINT@101,"LO SIENTO, SE ESTRELLO";
520 GOTO420
```



Antes de posarse en el suelo, el avión necesita construirse una pista de aterrizaje, destruyendo, para ello, todos los edificios que aparecen en la pantalla.



Si no encuentra sitio suficiente para aterrizar, o tropieza en el aire con el tejado de alguno de los rascacielos, el avión se estrella.



Si el piloto consigue un feliz aterrizaje el programa le felicita, informándole, además, sobre el número de bombas empleadas en la destrucción de todos los rascacielos.



ORDENADORES SOBRE RUEDAS

LAS tecnologías electrónica e informática han hecho su aparición en aviones, barcos y ferrocarriles desde hace casi más de una década. Por el contrario, el automóvil se ha visto relegado, hasta hace bien pocos años, al olvido electrónico.

Se dice que si la aviación hubiera progresado tanto como la electrónica, un Boeing podría dar hoy la vuelta al mundo en pocos minutos, consumiendo unos 20 litros de gasolina. Esta anécdota sirve para llamar la atención sobre una notable ausencia, la de la electrónica en el automóvil.

En los Estados Unidos diversas legislaciones «anti-contaminación», en riguroso vigor desde hace más de una década, han servido de incentivo para fabricantes de automóviles y fabricantes de semiconductores en el desarrollo de una nueva tecnología aplicada a motores y vehículos.

Por el contrario, en Europa las reglamentaciones no han sido tan estrictas. Pero otro acontecimiento no menos molesto ha venido a ocupar el espacio semivacío de las leyes protectoras del medio ambiente: el alza galopante en los precios para combustibles de automóviles. Esto, en definitiva, ha aumentado la preocupación de los fabricantes por conseguir que sus motores optimicen al máximo el consumo de carburante y, por consiguiente, reduzcan al mínimo el vertido de agentes contaminantes por los tubos de escape que, al fin, proceden en su mayoría de una combustión deficiente en el interior de los cilindros.

Economía electrónica

Con todo, la aplicación de la microelectrónica en el automóvil no se traduce exclusivamente en un menor consumo y contaminación. La electrónica puede ayudar a una reducción de costes sustancial, al permitir el ahorro de muchos metros de cableado. Por ejemplo, en lugar de tener un circuito para cada intermitente, basta con un mismo hilo que conecte a los cuatro: la electrónica se encarga de hacer funcionar al adecuado. De esta manera puede reducirse el cableado total de un automóvil hasta en un 85 por 100. Y esto puede ser importante; según un estudio del Real Automóvil Club británico, más del 40 por 100 de las averías que se producen en los coches tienen su causa en un fallo eléctrico.

Diagnóstico por enchufe

Otra de las ventajas de la aplicación de sistemas electrónicos en el automóvil es la reducción de costes de mantenimiento. El ordenador permite centralizar toda la información sobre el estado del motor, así como de todos los elementos mecánicos y eléctricos que in-

Los salpicaderos de los automóviles del futuro serán monitores de vídeo que informarán al piloto de todos los datos necesarios para una conducción más segura.

ORDENADORES SOBRE RUEDAS

corpora un automóvil moderno. A partir de aquí, las revisiones —tanto si son provocadas por una avería como si se trata de un repaso periódico— se reducen a testear la memoria del ordenador, operación que puede llevarse a cabo en pocos minutos y por el simple método de «volcar» toda la información contenida en sus memorias en el ordenador del taller mecánico, vía interface.

Una lámpara fundida, una bujía que ha reducido su rendimiento o incluso un fallo oculto en los sistemas «testigo» de elementos vitales, son recogidos por la unidad central y almacenados en memoria para su reparación, o bien comunicados inmediatamente al conductor. El estado de los frenos, el nivel de líquido en el circuito de frenado y en el del lavalunas, junto con los indicadores de cierre de las puertas, son controlados por el ordenador: un mecánico que jamás tiene descuidos.

Con todo, cabe hacerse una pregunta: ¿qué sucede si se avería el ordenador de a bordo? El sistema, en ese caso, está programado para «puentear» la unidad central y alimentar el motor bajo unos parámetros mínimos que permitan llegar hasta el taller más próximo.

Seguridad VLSI

Pero la electrónica aplicada al automóvil aporta la mayor satisfacción que se puede esperar de cualquier esfuerzo de un fabricante sobre sus vehículos: el aumento de la seguridad en ruta.

Cualquiera puede aceptar que un incremento en el volumen y calidad de la información proporcionada por los sistemas de control del automóvil al conductor, lleva aparejado un mayor coeficiente de seguridad. Y es precisamente la seguridad, al margen de otros logros técnicos nada despreciables, lo que los automovilistas de hoy reclaman cada vez con mayor firmeza. De poco sirve tener el vehículo más veloz o potente del mercado si una rueda baja de presión puede ocasionar un accidente y, posiblemente, daños irreparables en los ocupantes.

En este sentido cabe señalar el esfuerzo de algunas firmas de semiconductores en la investigación de nuevos sistemas de seguridad, tales como frenos antibloqueo, suspensión autocontrolada o sistemas de control para la caja de cambios.

Asimismo, el conductor de un futuro no lejano podrá abrir y cerrar las puertas de su vehículo por control remoto, leer en la pantalla del salpicadero mensajes de las autoridades de tráfico sobre el estado de las carreteras, o bien dejar que sea el propio ordenador el que mantenga la velocidad de crucero en un viaje de larga distancia, así como el control de la calefacción y la selección de su música favorita.



El ordenador de a bordo del BMW ejecuta diez funciones que van desde el control de la temperatura exterior, hasta las de ahorro de gasolina, pasando por la creación y almacenamiento de un código antirrobo.